UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/713,733	11/13/2003	Elmootazbellah Nabil Elnozahy	AUS920030760US1	2697

45992 7590 03/24/2009

IBM CORPORATION (JVM)
C/O LAW OFFICE OF JACK V. MUSGROVE
2911 BRIONA WOOD LANE
CEDAR PARK, TX 78613

EXAMINER	
SAVLA, ARPAN P	

ART UNIT	PAPER NUMBER
2185	

MAIL DATE	DELIVERY MODE
03/24/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/713,733
Filing Date: November 13, 2003
Appellant(s): ELNOZAHY ET AL.

Jack V. Musgrove
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed November 11, 2008 appealing from the Office action mailed June 12, 2008.

The appeal brief is filed in the new format under the revised BPAI final rule before the effective date of the BPAI final rule. The Office published the BPAI final rule to amend the rules governing practice before the BPAI in *ex parte* patent appeals. *See Rules of Practice Before the Board of Patent Appeals and Interferences in Ex Parte Appeals; Final Rule, 73 FR 32938 (June 10, 2008), 1332 Off. Gaz. Pat. Office 47 (July 1, 2008).* However, the effective date for the BPAI final rule has been delayed. *See Rules of Practice Before the Board of Patent Appeals and Interferences in Ex Parte Appeals; Delay of Effective and Applicability Dates, 73 FR 74972 (December 10, 2008).* In the notice published on November 20, 2008, the Office indicated that the Office will not hold an appeal brief as non-compliant solely for following the new format even though it is filed before the effective date. *See Clarification of the Effective Date Provision in the Final Rule for Ex Parte Appeals, 73 FR 70282 (November 20, 2008).* Since the appeal brief is otherwise acceptable, the Office has accepted the appeal brief filed by appellant.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,434,681	Armangau	8-2002
6,175,906	Christie	1-2001
6,725,289	Waldspurger et al.	4-2004
5,974,507	Arimilli et al.	10-1999

Applicant's admitted prior art (AAPA).

Romer et al. "Reducing TLB and Memory Overhead Using Online Superpage Promotion", 22-24 June 1995, 22nd Annual International Symposium on Computer Architecture, pp. 176-187.

Talluri et al. "Surpassing the TLB performance of superpages with less operating system support", 1994, ACM SIGOPS Operating Systems Review, Volume 28, Issue 5.

Hennessy, John L., Patterson, David A., "Computer Organization and Design, The Hardware/Software Interface", 1998, Morgan Kaufmann Publishers, Inc., Second Edition, pp. 657, 658, 668.

Tanenbaum, Andrew S., "Structured Computer Organization", 1984, Prentice-Hall, Inc., 2nd Edition, pp. 10-12.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1-5, 7, 9-11, 14, 17, and 19-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicant's "Description of the Related Art", hereafter "Applicant's admitted prior art (AAPA)" in view of Armangau (U.S. Patent 6,434,681).

As per claim 1, AAPA discloses a method of assigning virtual memory to physical memory in a data processing system, comprising the steps of:

allocating a set of physical memory pages of the data processing system for a new virtual superpage mapping (pg. 5, lines 13-16);

instructing a memory controller of the data processing system to move a plurality of virtual memory pages corresponding to an old page mapping to the set of physical memory pages corresponding to the new virtual superpage mapping (pg. 5, lines 16-17). *It should be noted that the "processor" is analogous to the "memory controller."*

AAPA does not expressly disclose accessing at least one of the virtual memory pages using the new virtual superpage mapping while the memory controller is copying old physical memory pages to new physical memory pages.

Armangau discloses accessing at least one of the virtual memory pages using the new virtual page mapping while the memory controller is copying old physical memory pages to new physical memory pages (col. 2, lines 16-18; col. 15, line 52 – col. 16, line 13; Fig. 7B, elements 124-129). *It should be noted that the "snapshot volume" is analogous to the "new virtual page mapping", the "production volume" is analogous to the "old physical memory pages", and the read/write access during snapshot maintenance is analogous to access operations while copying.*

AAPA and Armangau are analogous art because they are from the same field of endeavor, that being memory mapping systems.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to implement the mapping of Armangau's snapshot volume as AAPA's superpage mapping.

The motivation for doing so would have been to reduce delay when host write access to storage locations containing original data is delayed until the original data are transmitted to a backup storage device by providing a snapshot facility in the data storage system (Armangau, col. 2, lines 4-9).

Therefore, it would have been obvious to combine AAPA and Armangau for the benefit of obtaining the invention as specified in claim 1.

As per claim 2, the combination of AAPA/Armangau discloses said allocating step allocates a contiguous set of physical memory pages (AAPA, pg. 5, lines 14-16).

As per claim 3, the combination of AAPA/Armangau discloses said accessing step includes the step of directing a read operation for an address of the new page mapping which is currently being copied to a corresponding address of an old page mapping (Armangau, col. 2, lines 16-18).

As per claim 4, the combination of AAPA/Armangau discloses said accessing step includes the step of directing a write operation for an address of the new page mapping which is currently being copied to both the address of the new page mapping and a corresponding address of an old page mapping (Armangau, col. 15, line 52 – col. 16, line 13; Fig. 7B, elements 124-129).

As per claim 5, the combination of AAPA/Armangau discloses said accessing step includes the step of directing a write operation for an address of the new page mapping which has not yet been copied to a corresponding address of an old page mapping (Armangau, col. 15, lines 43-51; Fig. 7B, elements 122 and 123).

As per claim 7, AAPA discloses a memory controller comprising:

an input for receiving remapping instructions for a virtual superpage (pg. 5, lines 16-17).

AAPA does not expressly disclose a mapping table which temporarily stores entries of old page addresses and corresponding new page addresses associated with the page remapping instructions;

and a memory access device which directs the copying of memory pages from the old page addresses to the new page addresses while handling access operations which use the new page addressees, and releases the entries in said mapping table as copying for each entry is completed.

Armangau discloses a mapping table which temporarily stores entries of old page addresses and corresponding new page addresses associated with the page remapping instructions (col. 6, lines 42-48; Fig. 1, elements 20, 21, and 24); *It should be noted that the "backup command" is analogous to "remapping instructions."*

and a memory access device which directs the copying of memory pages from the old page addresses to the new page addresses while handling access operations which use the new page addressees, and releases the entries in said mapping table as copying for each entry is completed (col. 2, lines 16-18; col. 15, line 52 – col. 16, line 13; Fig. 7B, elements 124-129; col. 6, lines 42-50; col. 7, line 65 – col. 8, line 3; Fig. 1, element 21; Fig. 3, element 51). *It should be noted that the "storage controller" within the "primary data storage subsystem" is analogous to the "memory access device."*

AAPA and Armangau are analogous art because they are from the same field of endeavor, that being memory mapping systems.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to implement the mapping of Armangau's snapshot volume as AAPA's superpage mapping.

The motivation for doing so would have been to reduce delay when host write access to storage locations containing original data is delayed until the original data are

transmitted to a backup storage device by providing a snapshot facility in the data storage system (Armangau, col. 2, lines 4-9).

Therefore, it would have been obvious to combine AAPA and Armangau for the benefit of obtaining the invention as specified in claim 7.

As per claim 9, the combination of AAPA/Armangau discloses said memory access device directs a read operation for a new page address which is currently being copied to a corresponding old page address (Armangau, col. 2, lines 16-18).

As per claim 10, the combination of AAPA/Armangau discloses said memory access device directs a write operation for a new page address which is currently being copied to both the new page address and a corresponding old page address (Armangau, col. 15, line 52 – col. 16, line 13; Fig. 7B, elements 124-129).

As per claim 11, the combination of AAPA/Armangau discloses said memory access device directs a write operation for a new page address which has not yet been copied to a corresponding old page address (Armangau, col. 15, lines 43-51; Fig. 7B, elements 122 and 123).

As per claim 14, AAPA discloses a computer system comprising:

a new virtual superpage mapping (pg. 5, lines 13-16).

AAPA does not expressly disclose a processing unit;

an interconnect bus connected to said processing unit;

a memory array;

and a memory controller connected to said interconnect bus and said memory array, wherein said memory controller copies memory pages from old page addresses to new page addresses according to a new virtual superpage mapping while handling access operations which use the new page addresses and while said processing unit carries out program instructions using the new page addresses.

Armangau discloses a processing unit (col. 6, lines 1-2; Fig. 1, element 20);

an interconnect bus connected to said processing unit (Fig. 1, the "line" (i.e. bus) between the host the primary data storage subsystem)

a memory array (col. 6, lines 3-6; Fig. 1, element 27);

and a memory controller connected to said interconnect bus and said memory array, wherein said memory controller copies memory pages from old page addresses to new page addresses according to a new virtual superpage mapping while handling access operations which use the new page addresses and while said processing unit carries out program instructions using the new page addresses (col. 2, lines 16-18; col. 15, line 52 – col. 16, line 13; Fig. 7B, elements 124-129; col. 6, lines 42-50; Fig. 1, element 21; Fig. 3, element 51). *It should be noted that the "storage controller" within the "primary storage subsystem" is analogous to the "memory controller."*

AAPA and Armangau are analogous art because they are from the same field of endeavor, that being memory mapping systems.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to implement the mapping of Armangau's snapshot volume as AAPA's superpage mapping.

The motivation for doing so would have been to reduce delay when host write access to storage locations containing original data is delayed until the original data are transmitted to a backup storage device by providing a snapshot facility in the data storage system (Armangau, col. 2, lines 4-9).

Therefore, it would have been obvious to combine AAPA and Armangau for the benefit of obtaining the invention as specified in claim 14.

As per claim 17, the combination of AAPA/Armangau discloses said cache modifies the address tag of the cache entry in response to a determination that the cache entry contains a valid value which is not present elsewhere in the system (Armangau, col. 10, lines 50-67).

As per claim 19, the combination of AAPA/Armangau discloses said memory controller includes:

a mapping table which temporarily stores entries of old page addresses and corresponding new page addresses (Armangau, col. 7, lines 18-20; Fig. 1, element 26); *See the citation note for the similar limitation in claim 7 above.*

and a memory access device which directs the copying of the memory pages from the old page addresses to the new page addresses and releases the entries in

said mapping table as copying for each entry is completed (Armangau, col. 6, lines 42-50; col. 7, line 65 – col. 8, line 3; Fig. 1, element 21). *See the citation note for the similar limitation in claim 7 above.*

As per claim 20, the combination of AAPA/Armangau discloses said processing unit, said interconnect bus, said memory array and said memory controller are all part of a first processing cluster, and further comprising a network interface which allows said first processing cluster to communicate with a second processing cluster, said memory controller having at least one pointer for a new page address which maps to a physical memory location in said second processing cluster (Armangau, col. 6, lines 1-3; Fig. 1, element 22). *It should be noted that the "second storage subsystem" is analogous to the "second processing cluster." It should also be noted that it is inherently required the second storage subsystem include some sort of "interface" in order to communicate with the first storage subsystem.*

Claims 6 and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA in view of Armangau as applied to claims 1 and 14 above, and further in view of Christie (U.S. Patent 6,175,906).

As per claim 6, the combination of AAPA/Armangau discloses all the limitations of claim 6 except the step of updating an entry in a cache memory of the data

processing system which corresponds to a memory location in the virtual memory page, by modifying an address tag of the cache entry according to the new page mapping.

Christie discloses the step of updating an entry in a cache memory of the data processing system which corresponds to a memory location in the virtual memory page, by modifying an address tag of the cache entry according to the new page mapping (col. 4, lines 40-50). *It should be noted that "invalidating" the tags is analogous to "modifying" the tags.*

The combination of AAPA/Armangau and Christie are analogous art because they are from the same field of endeavor, that being memory systems.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to implement Christie's revalidation of virtual tags within AAPA/Armangau's storage system because all the claimed elements were known in the prior art and one skilled in the art could have combined the elements as claimed by known methods with no change in their respective functions, and the combination would have yielded the predictable results of increasing speed of recovery from remapping by reducing the number of TLB accesses during virtual-to-physical memory remapping.

Therefore, it would have been obvious to combine AAPA/Armangau and Christie for the benefit of obtaining the invention as specified in claim 6.

As per claim 16, the combination of AAPA/Armangau/Christie discloses said processing unit has a processor core and an associated cache (Armangau, Fig. 1, element 21; Fig. 3, element 52);

and said cache modifies an address tag of a cache entry which corresponds to a memory location in the new page addresses (Christie, col. 4, lines 40-50).

Claim 8 is rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA in view of Armangau as applied to claim 7 above, and further in view of Romer et al. "Reducing TLB and Memory Overhead Using Online Superpage Promotion", (hereinafter "Romer").

As per claim 8, the combination of AAPA/Armangau discloses all the limitations of claim 8 except said mapping table has 32 slots for receiving corresponding pairs of the old page addresses and new page addresses.

Romer discloses said mapping table has 32 slots for receiving corresponding pairs of the old page addresses and new page addresses (pg. 178, italicized section entitled "Table 2", line 4). *It should be noted that the "entries" is analogous to the "slots."*

The combination of AAPA/Armangau and Romer are analogous art because they are from the same field of endeavor, that being memory systems.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to implement Romer's 32 slot TLB as AAPA/Armangau's primary directory.

The motivation for doing so would have been to improve system performance by increasing instructions per TLB miss (Romer, pg. 187, section entitled "Capacity Counters", last paragraph).

Therefore, it would have been obvious to combine AAPA/Armangau and Romer for the benefit of obtaining the invention as specified in claim 8.

Claims 12 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA in view of Armangau as applied to claim 7 above, and further in view of Waldspurger et al. (U.S. Patent 6,725,289) (hereinafter "Waldspurger").

As per claim 12, thr combination of AAPA/Armangau discloses all the limitations of claim 12 except said memory access device includes a state engine which sequentially reads the paired old and new pages addresses in said mapping table.

Waldspurger discloses said memory access device includes a state engine which sequentially reads the paired old and new pages addresses in said mapping table (col. 7, lines 26-36; col. 11, lines 13-21 and 52-62; Fig. 2, element 610). *It should be noted that the "mapping module" is analogous to the "state engine" and that "page numbers" are analogous to "page addresses." It should also be noted that the mapping module reads both the old page numbers and new page numbers because it reads the page numbers before the remapping (i.e. the old or original page numbers) and also reads the page numbers after remapping (i.e. the new page numbers).*

The combination of AAPA/Armangau and Waldspurger are analogous art because they are from the same field of endeavor, that being memory systems.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to implement Waldspurger's transparent address remapping within AAPA/Armangau's storage system because all the claimed elements were known in the prior art and one skilled in the art could have combined the elements as claimed by known methods with no change in their respective functions, and the combination would have yielded the predictable results of remapping non-contiguous pages to contiguous pages, thus avoiding the need for page splits on future transfers which will improve performance accordingly.

Therefore, it would have been obvious to combine AAPA/Armangau and Waldspurger for the benefit of obtaining the invention as specified in claim 12.

As per claim 13, the combination of AAPA/Armangau/Waldspurger discloses said memory access device further includes a direct memory access (DMA) engine controlled by said state engine which carries out actual copying of the memory pages (Waldspurger, col. 7, line 67 – col. 8, line 3; col. 8, lines 20-26; Fig. 2, element 116). *It should be noted that the "memory management unit (MMU)" is analogous to the "DMA engine."*

Claim 15 is rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA in view of Armangau as applied to claim 14 above, and further in view of

Talluri et al. "Surpassing the TLB Performance of Superpages with Less Operating System Support" (hereinafter "Talluri").

As per claim 15, the combination of AAPA/Armangau discloses copying of memory pages by a memory controller (Armangau, col. 2, lines 16-18; col. 15, line 52 – col. 16, line 13; Fig. 7B, elements 124-129; col. 6, lines 42-50; Fig. 1, element 21; Fig. 3, element 51).

The combination of AAPA/Armangau does not disclose said processing unit includes a processor core having a translation lookaside buffer (TLB) whose entries keep track of current virtual-to-physical memory address assignments;

and said TLB entries are updated for the new page addresses prior to completion of copying of the memory pages by the memory.

Talluri discloses said processing unit includes a processor core having a translation lookaside buffer (TLB) whose entries keep track of current virtual-to-physical memory address assignments (pg. 1, right column, first full paragraph);

and said TLB entries are updated for the new page addresses prior to completion of copying of the memory pages by the memory (pg. 3, right column, second full paragraph). *It should be noted that a "page promotion" process itself is the copying of memory pages.*

The combination of AAPA/Armangau and Talluri are analogous art because they are from the same field of endeavor, that being memory systems.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to implement Talluri's subblock TLB within AAPA/Armangau's storage system because all the claimed elements were known in the prior art and one skilled in the art could have combined the elements as claimed by known methods with no change in their respective functions, and the combination would have yielded the predictable results of increasing system speed by improving performance of TLBs in a superpage environment.

Therefore, it would have been obvious to combine AAPA/Armangau and Talluri for the benefit of obtaining the invention as specified in claim 15.

Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA in view of Armangau as applied to claim 16 above, and further in view of Arimilli et al. (U.S. Patent 5,974,507) (hereinafter "Arimilli").

As per claim 18, the combination of AAPA/Armangau discloses all the limitations of claim 18 except said cache further relocates the cache entry based on a changed congruence class for the modified address tag.

Arimilli discloses said cache further relocates the cache entry based on a changed congruence class for the modified address tag (col. 6, lines 43-66). *It should be noted that the intermittent or real-time adjustment of the members (i.e. cache entries) of the congruency class is based on the modification of address bits of the cache entries.*

The combination of AAPA/Armangau and Arimilli are analogous art because they are from the same field of endeavor, that being memory systems.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to implement Arimilli's programmable congruence class caching mechanism within AAPA/Armangau's storage system because all the claimed elements were known in the prior art and one skilled in the art could have combined the elements as claimed by known methods with no change in their respective functions, and the combination would have yielded the predictable results of improving cache efficiency by lessening eviction rates.

Therefore, it would have been obvious to combine AAPA/Armangau and Arimilli for the benefit of obtaining the invention as specified in claim 18.

(10) Response to Argument

Rejection of claims 1-5, 7, 9-11, 14, 17, 19, and 20 under §103(a)

Appellant argues on pages 10-11 of the Appeal Brief that:

"The first error in the Office Action is the assertion that a "processor" is the same as a memory controller".

The Examiner respectfully disagrees. Firstly, the Examiner reasserts that the terms processor and memory controller are terms used interchangeably within the art. In many modern CPU designs, the memory controller is in fact located on the processor die itself. Appellant has even admitted that a processor is involved with access to

memory systems (see the second full paragraph on page 2 of the communication filed July 17, 2007). Notwithstanding, the Examiner offers John L. Hennessy and David A. Patterson, "Computer Organization and Design, The Hardware/Software Interface, Second Edition", pages 657, 658, and 668 as extrinsic evidence. As can be clearly seen from Figures 8.7, 8.8, and 8.12, it is in fact the processor that manages the memory array and it is the processor that executes memory data reads and memory data writes (emphasis added). Therefore, Appellant's narrow characterization of the functionalities of "processors" and "memory controllers" (see Appeal Brief, pg. 11) is clearly in error.

Secondly, and even more importantly, when looking at the claims themselves, Appellant's "memory controller" merely moves (i.e. copies) memory pages corresponding to an old (i.e. original) page mapping to a new superpage mapping. AAPA's "processor" copies memory pages from an original mapping to a new superpage mapping (see Appellant's specification, pg. 5, lines 16-17). Thus, even though the names used to label certain elements in AAPA and Appellant's invention may differ, when simply looking at the claim language to see the actual functionality provided by Appellant's "memory controller", it is evident that AAPA's "processor" discloses the same functions carried out by Appellant's "memory controller", as simply and broadly claimed. Accordingly, AAPA's "processor" is functionally equivalent to Appellant's "memory controller" and therefore AAPA's "processor" anticipates Appellant's "memory controller."

Appellant argues on pages 11-13 of the Appeal Brief that:

"The second error in the Office Action is the implication that the prior art software procedure for superpage remapping is analogous to Appellant's claimed hardware-based method."

The Examiner respectfully disagrees. The Examiner notes that the features upon which Appellant relies (i.e., "hardware") are not recited anywhere in the rejected claims. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Appellant's argument concerning the distinction between software supervision of page remapping and hardware-based control is not commensurate with the language of the claims because nowhere in any of the rejected claims does the word "hardware" even appear.

Assuming, *in arguendo*, that the limitation "hardware" was recited in the claims, such a limitation is not a patentable distinction because it is well known in the art that hardware and software are logically equivalent. The Examiner offers Andrew S. Tanenbaum, "Structured Computer Organization, 2nd Edition", pages 10-12 as extrinsic evidence which shows that hardware and software are logically equivalent.

Appellant argues on pages 13-15 of the Appeal Brief that:

"The third error in the Office Action is the assertion that Armangau accesses a virtual superpage using a new mapping while the memory controller is still copying pages."

The Examiner respectfully disagrees. The Examiner would notes that Appellant's argue that their "new page mapping" involves "coalescing" and that the new superpage is "different" and "more efficient", however, these features upon which Appellant relies (i.e., "coalescing", "different", and "more efficient") are not recited in the rejected claims. Again, although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Thus, based on the actual claim language there is no change in "efficiency" between the old page mapping and the new page mapping as Appellant alleges.

Additionally, the Examiner notes that Appellant is attacking the references individually. One cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). The combination of AAPA/Armangau does in fact disclose a virtual superpage mapping which is "new" because the snapshot mapping is not the exact same as the original (i.e. old) production volume due to write operations completed during the copying process. Accordingly, when taking the broadest reasonable interpretation of the limitation "new virtual superpage mapping", the combination of AAPA/Armangau sufficiently disclose said limitation. Based on the foregoing, the combination of AAPA/Armangau renders claims 1-5, 7, 9-11, 14, 17, 19, and 20 unpatentable.

(i) Claims 3 and 9

The Examiner respectfully disagrees with Appellant's argument regarding claims 3 and 9 on page 16 of the Appeal Brief. Col. 15, line 52 – col. 16, line 13, Fig. 7B, elements 124-129 of Armangau clearly show that during the copying process, when a write is directed to the snapshot volume, the track is written to both the snapshot volume as well as the production volume. Thus, the snapshot volume (i.e. new mapping) is accessed while the copying of data from the production volume to the snapshot volume is taking place. Accordingly, the combination of AAPA/Armangau renders claims 3 and 9 unpatentable.

Rejection of claims 6 and 16 under §103(a)

The Examiner respectfully disagrees with Appellant's argument regarding claims 6 and 16 on page 17 of the Appeal Brief. Appellant argues that "deassertion of a valid bit in the valid register does not remap an address which is in the virtual tag register. This feature of Appellants' invention allows the superpage construction to complete without having to flush the contents of any affected cache memory" (see Appeal Brief, pg. 17), however, the Examiner notes that neither the language of claims 6 or 16 calls for such a feature. Rather, claim 6 recites "...the step of updating an entry in a cache memory of the data processing system which corresponds to a memory location in the virtual memory page, by modifying an address tag of the cache entry according to the new page mapping" while claim 16 recites "...said cache modifies an address tag of a cache entry which corresponds to a memory location in the new page addresses."

(emphasis added) When taking the broadest reasonable interpretation of the terms "updating" and "modifying" it follows that the Christie's step of "invalidating" tags sufficiently reads on "updating" and/or "modifying" tags. Thus, even though the virtual address tags and valid bits are in different registers in Christie, the act of "invalidating" a virtual address tag (which is done by deasserting a valid bit) discloses "updating/modifying" an address tag, as simply and broadly claimed by Appellant. Accordingly, the combination of AAPA/Armangau/Christie renders claims 6 and 16 unpatentable.

Rejection of claim 8 under §103(a)

The Examiner respectfully disagrees with Appellant's argument regarding claim 8 on page 18 of the Appeal Brief. As addressed in the section titled "Rejection of claims 1-5, 7, 9-11, 14, 17, 19, and 20 under §103(a)" above, the combination of AAPA/Armangau discloses using new virtual superpage mappings while copying of the underlying memory pages is still ongoing. Claim 8 further recites said mapping table has 32 slots for receiving corresponding pairs of the old page addresses and new page addresses. As detailed in the rejection above, Romer discloses said mapping table has 32 slots for receiving corresponding pairs of the old page addresses and new page addresses (pg. 178, italicized section entitled "Table 2", line 4). Accordingly, the combination of AAPA/Armangau/Romer renders claim 8 unpatentable.

Rejection of claims 12 and 13 under §103(a)

The Examiner respectfully disagrees with Appellant's argument regarding claims 12 and 13 on pages 18-19 of the Appeal Brief. As addressed in the section titled "Rejection of claims 1-5, 7, 9-11, 14, 17, 19, and 20 under §103(a)" above, the Examiner notes that the features upon which Appellant relies (i.e., "hardware") are not recited anywhere in the rejected claims. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Appellant's argument concerning the distinction between a software mapping module and a hardware state machine is not commensurate with the language of the claims because nowhere in any of the rejected claims does the word "hardware" even appear. Accordingly, the combination of AAPA/Armangau/Waldspurger renders claims 12-13 unpatentable.

(i) Claim 13

The Examiner respectfully disagrees with Appellant's argument regarding claim 13 on page 19 of the Appeal Brief. The Examiner notes that the cited section of Waldspurger used in the rejection of claim 13 above (col. 7, line 67 – col. 8, line 3; col. 8, lines 20-26; Fig. 2, element 116) discloses the translation of VPNs to PPNs done by memory management unit (MMU) 116. Col. 6, lines 34-37 of Waldspurger further disclose "The actual translation may be accomplished simply by replacing the VPN (the higher order bits of the virtual address) with its PPN mapping, leaving the lower order

offset bits the same." Thus, the step of "replacing" the VPN with the PPN by the MMU discloses Appellent's step of "copying" by the DMA engine, as simply and broadly claimed. Accordingly, the combination of AAPA/Armangau/Waldspurger renders claim 13 unpatentable.

Rejection of claim 15 under §103(a)

The Examiner respectfully disagrees with Appellant's argument regarding claim 15 on page 20 of the Appeal Brief. As stated in pg. 3, right column, second full paragraph of Talluri, "*Page promotion* is the mechanism by which a set of pages are coalesced to a larger superpage." Hence, the page promotion process itself is the copying of memory pages. As is also evident from the cited portion of Talluri, the page promotion process involves updating page tables and TLBs. Since updating page tables and TLBs is part of the page promotion process itself, it follows that it is logically impossible for the updating of page tables and TLBs to occur after the page promotion process completes (because without the updating page tables and TLBs the page promotion process would never actually complete). Rather, the updating of page tables and TLBs must be done at some finite time after the page promotion process begins but before the page promotion process completes. Thus, in Talluri the TLB entries ("pages tables and TLBs") are updated for the new page addresses prior to completion of copying of the memory pages ("the page promotion process"). Accordingly, the combination of AAPA/Armangau/Talluri renders claim 13 unpatentable.

Rejection of claim 18 under §103(a)

The Examiner respectfully disagrees with Appellant's argument regarding claim 18 on pages 20-21 of the Appeal Brief. As addressed in the section titled "Rejection of claims 1-5, 7, 9-11, 14, 17, 19, and 20 under §103(a)" above, the combination of AAPA/Armangau discloses using new virtual superpage mappings while copying of the underlying memory pages is still ongoing. The Arimilli reference deals with cache eviction based on congruence class. As is evident from the portion of Arimilli cited in the rejection above, Arimilli discloses the ability to program (i.e. change) congruence classes of cache blocks. With respect to the instant invention, the Examiner notes that the claim language does not specify where exactly the cache entry is relocated to. Thus, when taking the broadest reasonable interpretation of the term "relocate" it follows that Arimilli's "eviction" of a cache block from cache (i.e. one location) to main memory (i.e. another location) sufficiently reads on "relocating" a cache entry. Thus, Arimilli discloses relocating ("evicting") a cache entry based on a changed ("reprogrammed") congruence class for the modified address tag ("address bits"). Accordingly, the combination of AAPA/Armangau/Arimilli renders claim 18 unpatentable.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Arpan Savla/

Examiner, Art Unit 2185

/Sanjiv Shah/

Supervisory Patent Examiner, Art Unit 2185

/Manorama Padmanabhan/

Quality Assurance Specialist, TC2100